

target application through said communication network for obtaining meta-information in respect of target application.

5 6. An automatic test system as claimed in claim 1 wherein said target application includes a downloading means for installing reflection objects received from said test generation means.

10 7. An automatic test system as claimed in claim 1 wherein said target application on target device contains reflection objects for downloading meta-information to said test generation means through said communication network.

15 8. An automatic test system as claimed in claim 1 wherein said target application operates under an environment which supports reflection viz. the AperiOS operating system (Sony's Realtime OS) or is in Java.

20 9. An automatic test system as claimed in claim 1 wherein said test generation means also includes a means for generating test cases independently of API or methods for which the test cases are generated.

25 10. An automatic test system as claimed in claim 1 wherein said test generation means further comprises a configuration module, test design module, test driver module, test execution module and a report module, all connected to the said data storage means through said network.

11. An automatic test system as claimed in claim 10 wherein the configuration module is a software executing on a computing system, which obtains information on test techniques, object details and data type details from the user for defining the test cases.

12. An automatic test system as claimed in claim 10 wherein the test design module is a software executing on a computing system which provides the test scenario framework to create test scenarios and the information stored in said data storage means.

13. An automatic test system as claimed in claim 10 wherein the test driver module is a software executing on a computing system, which automatically generates the test cases and then the test programs in a description language using the test scenario provided by the said test design module and the information in said data storage means.

14. An automatic test system as claimed in claim 10 wherein the test execution module loads the image created by the said image builder module on said target application and monitors and controls the execution of image on said target application.

15. An automatic test system as claimed in claim 1 wherein the image builder means is a software executing on a computing system, which converts the test program received from the said test driver module in description language to an image form suitable for loading and executing on the said target application.

16. An automatic test system as claimed in claim 10 wherein the said report module is a software executing on a computing system for generating reports from the results of the testing on the target application, which are stored in the said data storage means.

17. An automatic test system as claimed in claim 1 wherein the said data storage means is a software executing on a computing system for storing information relating to test scenario, test technique, object details, results of tests and incorporates object serialization means in order to improve time for execution and improve security.

18. An automatic test system as claimed in claim 1 wherein said test generation means is developed in Java making it hardware and software independent and the test program generated is in DL (description language).

19. An automatic test system as claimed in claim 18 wherein said description language is Standard Description Language (SDL)

20. An automatic test system as claimed in claim 18 wherein the said description language is converted by an appropriate language code converter to the desired test language.

21. An automatic test system as claimed in claim 20 wherein the said code converter to convert the description language test program to the desired language test program is provided either at test driver module of the said test generation means or at the image builder means.

22. An automatic test system as claimed in claim 1 wherein the said data storage means is a server and is connected through ODBC / JDBC and is not dependent on any particular database, the said server is also developed in Java making it hardware and software independent using object serialization for communication.

23. An automatic test system as claimed in claim 1 wherein the image builder means consists of an appropriate compiler and linker to generate an executable data image.
- 5 24. An automatic test system as claimed in claim 1 wherein the said test generation means further includes a means for simultaneously testing a plurality of target applications at one location or at multiple locations.
- 10 25. An automatic test system as claimed in claim 1 wherein a fire-wall is either provided between said test generation means and said communication network or between the said communication network and said target applications or at both places for access control of communication.
- 15 26. An automatic test system as claimed in claim 1 wherein the said communication network comprising LAN, IEEE 1394 network or internet, wireless communication network, FTTH (Fiber To The Home), CATV, or xDigital Subscriber Line (xDSL).
- 20 27. An automatic test system as claimed in claim 1 wherein the target application is a set of software.
28. An automatic test system as claimed in claim 1 wherein the said target device is used for running one or more target applications.
- 25 29. A method for testing remote target applications comprising the steps of:
- obtaining meta-information details of the target application,
 - checking the said meta-information against the stored meta-information,

- updating the stored meta-information in case of discrepancy or absence of the obtained meta-information,
- automatically generating test cases based on said meta-information,
- creating test scenario and generating test cases from the designed test scenario,
- adding or modifying the said test cases by user input,
- automatically generating test program using the test scenario and the test cases,
- building the test image from the said test program,
- downloading said test image to said target application for testing,
- getting information from the user (test engineer) with regard to the order of execution, repetition and resetting of target application,
- automatically testing the target application,
- generating the reports from the test results in a required format.

30. A method as claimed in claim 29 wherein the meta-information details of the target application are obtained using the reflection principle either by the use of reflection object bundled with the target application or by downloading the reflection object to the target application.

31. A method as claimed in claim 29 wherein the test scenarios, test programs and test image are generated using object serialization in order to improve security of data communication over the network as well as to improve the utilization of resources in the network in order to reduce time for execution.

32. A method as claimed in claim 29 wherein the said test programs are generated independent of the API or the method for which they are applicable.

33. A method as claimed in claim 29 wherein the said test program is generated by:

- providing the framework to define the test scenarios using said meta-information,
- generating different possible test cases automatically using said test scenarios,
- generating the test program in a description language using said test scenarios and test cases.

34. A method as claimed in claim 29 wherein the execution of the test programs is conducted using the order of execution, the repetition, the requirement for resetting and batch information by user input.

35. A method as claimed in claim 29 wherein the reports are generated for the specified test scenarios.

36. A method as claimed in claim 29 wherein the solution is provided to a service station for testing the target application or the said service station is able to use the said automatic test system through a terminal provided at the service station.

37. A method as claimed in claim 29 wherein a plurality of target applications can be simultaneously tested either at one location or at multiple locations.